Amendments in the Claims

1. (currently amended) A computer-implemented method for identifying user interface (UI) objects in a markup-language stream, the method comprising the steps of:

receiving a predefined ~~application-specific~~ grammar for a particular application;

automatically generating a parser computer program based on the predefined ~~application-specific~~ grammar using an automated parser generator tool; ~~, the parser computer program for:~~

scanning ~~any of~~ the (i) the markup-language stream ~~and~~ or (ii) a corresponding document object model (DOM) with the parser computer program to generate tokens; ~~and~~

parsing the tokens with the parser computer program to identify ~~one or more~~ at least one UI objects in a portion of the particular application; and

outputting the portion of the particular application.

2. (original) The method of claim 1, wherein said markup-language stream drives a markup-language-based browser application, and wherein the scanning step includes scanning the DOM generated by a browser that displays that application.

3. (original) The method of claim 1, wherein the scanning step includes identifying elements of the DOM by traversal thereof.

4. (cancelled)

5. (currently amended) The method of claim 3, wherein the scanning step includes generating one or more tokens for each ~~parsed~~ scanned DOM element.

6. (cancelled) .

7. (cancelled).

8. (currently amended) The method of claim 7, wherein ~~said~~ the at least one UI objects comprises one of a user input fields, text fields, metatags, unprintable markup-language, ~~and~~ or an in-line images.

9. (original) The method of claim 1, wherein the scanning and parsing steps are adapted to identify UI objects that correspond to elements displayed in the markup-language application.

10. (currently amended) The method of claim 1~~9~~, ~~wherein the parser computer program is operable to~~ , further comprising grouping the tokens into syntactic structures that identify items displayed by the ~~markup-language~~ particular application.

11. (currently amended) The method of claim 10~~9~~, wherein said step of grouping comprises identifying similarly formatted markup-language elements based on their markup-language attributes such as classname, font size, style, tag color, and size.

12. (currently amended) The method of claim 1~~9~~, wherein said at least one UI objects comprises a name, content, a shape, or a location~~, and properties~~.

13. (previously presented) The method of claim 1, wherein automatically generating ~~said~~ the parser computer program comprises executing YACC ("Yet Another Compiler-Compiler").

14. (cancelled).

15. (cancelled).

16. (currently amended) The method of claim 1, wherein the parser computer program is a LALR(1) parser.

17. (currently amended) The method of claim 1, wherein the parser is a LR(1) parser.

18. (currently amended) The method of ~~any of~~ claims 1 ~~–17~~, wherein the markup language is any of HTML, XHTML and XUL.

19. (currently amended) A ~~digital data processing system for identifying user interface (UI) objects in markup language based applications~~ computer-readable medium encoded with computer program code, the computer program code comprising:

~~a parser computer program generator for execution on the digital data processing system to generate a parser computer program based on a predefined application specific grammar using an automated parser generator tool, the parser computer program to:~~

~~scan any of (i) a markup language stream and (ii) a corresponding document object model (DOM) to generate tokens; and~~

~~parse the tokens with the parser computer program to generate a list of UI objects.~~

program code for receiving a predefined grammar for a particular application;

program code for automatically generating a parser computer program based on the predefined grammar using an automated parser generator tool;

program code for scanning the (i) the markup-language stream or (ii) a corresponding document object model (DOM) with the parser computer program to generate tokens;

program code for parsing the tokens with the parser computer program to identify at least one UI objects in a portion of the particular application; and

program code for outputting the portion of the particular application.

20. (currently amended) The ~~system~~ computer-readable medium of claim 19, wherein the list of UI objects corresponds to elements displayed by the markup-language DOM.

21. (currently amended) The ~~system~~ computer-readable medium of claim 20, wherein said UI objects comprise name, content, shape, location, and properties.

22. (cancelled).

23. (currently amended) The ~~system~~ computer-readable medium of claim 19, wherein said tokens are interpreted according to the predefined grammar to identify and distinguish among UI objects of a markup-language application's display.

24. (currently amended) The ~~system~~ computer-readable medium of claim 19~~23~~, wherein the at least one UI objects comprises a user input fields, a text fields, a metatags, unprintable markup-language, ~~and~~ or an in-line images.

25. (previously presented) The ~~system~~ computer-readable medium ~~of any~~ of claims 19 - ~~24~~, wherein the markup language is any of HTML, XHTML and XUL.

26. (new) The method of claim 1, further comprising providing context-based help based at least in part on the portion of the particular application.

27. (new) The computer-readable medium of claim 19, further comprising program code for providing context-based help based at least in part on the portion of the particular application.